OSNEXUS Corporation

OSNEXUS Crypto Library
FIPS 140-2 Non-proprietary Security Policy

Document Revision: 1.0
S.W. Version: 1.0

REVISION HISTORY

Author(s)	Version	Updates
Steven Umbehocker	1.0	Initial Release

TABLE OF CONTENTS

1. INTRODUCTION

OSNEXUS Corporation, a Software Defined Storage Company, has developed a crypto library to support OSNEXUS QuantaStor applications (outside of scope of this validation). The cryptographic module under test is the **OSNEXUS Crypto Library, software version 1.0**, herein referred to as simply the module. The module isolates all crypto functionality used to manage encrypted storage into its boundary.

This document, the OSNEXUS Crypto Library FIPS 140-2 Non-proprietary Security Policy, herein referred to as the Security Policy, specifies the security rules under which the module must operate.

Detailed information on the module under validation is found below.
As per FIPS Implementation Guidance (Section G.5), the module will remain compliant in all operational environments for which the binary executable remains unchanged and INTEL RDRAND is supported by the CPU. The Cryptographic Module Validation Program (CMVP) makes no statement as to the correct operation of the module or the security strengths of the generated keys if the specific operational environment is not listed on the validation certificate.

Table 1 - Detailed Module Information

Module Name	OSNEXUS Crypto Library
Software Version	1.0
PAA capability	Supports both AES-NI Enabled and AES-NI Disabled
GPC	SuperMicro X10DRi motherboard (Server chassis SC216A-R900LPB)
Operational Environment	• Ubuntu Linux 16.04 running on X10DRi with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz with PAA • Ubuntu Linux 16.04 running on X10DRi with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz without PAA

2. SECURITY LEVEL SPECIFICATION

The module meets the overall security requirements of FIPS 140-2 for Level 1 software-only modules.

Table 2 - Module Security Level Specification

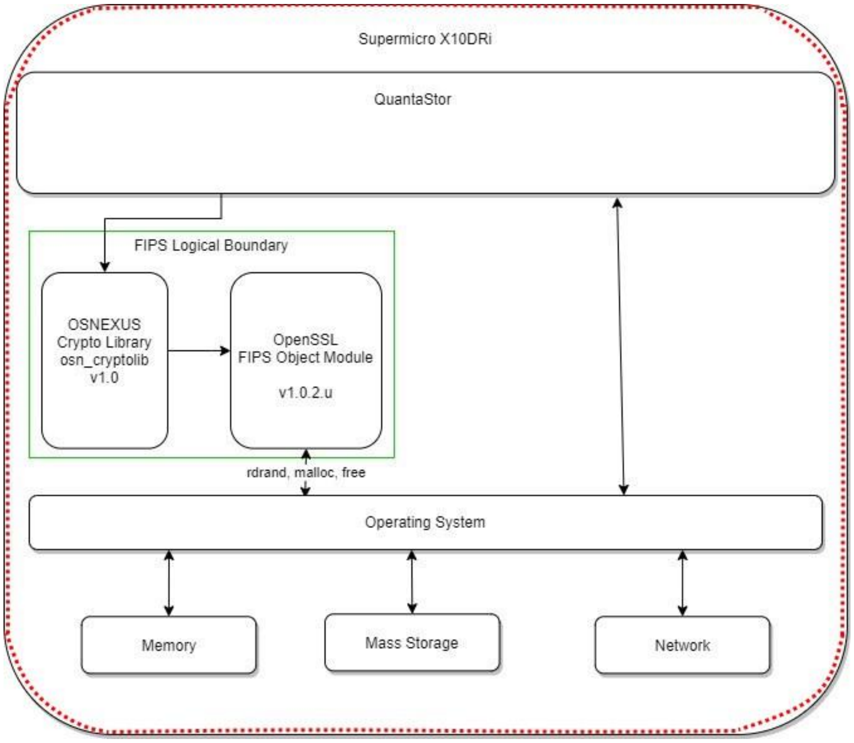
Security Requirements Area	**Level**
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-tests	1
Design Assurance	3
Mitigation of Other Attacks	N/A

3. MODULE SPECIFICATION & BOUNDARY

The module is a software-only module embodied as a shared library, which executes on General Purpose Computing (GPC) platforms. The physical cryptographic boundary of the module is the physical perimeter of the GPC under which the module has been installed. The logical cryptographic boundary of the module is defined as the logical perimeter encompassing the binary of the software library and the single exposed entry point (API) into the module. An operator is accessing the module whenever one of the library calls is executed through the API and thus the module logical interfaces are provided by the API calls.

For purposes of FIPS 140-2, the module is classified as a multi-chip standalone module.

Figure 1 - Figure of cryptographic boundary (red outline) and logical boundary (green outline)



4. PHYSICAL PORTS AND LOGICAL INTERFACES

The interface to the cryptographic module is via the module's API.

The Logical interfaces can be described as follows:

Table 3 - Description of logical interfaces

FIPS 140-2 Logical Interface	Description
Data Input	Data parameters passed to the module via API calls.
Data Output	Data returned by the module via API calls.
Control Input	Control Input – API function calls and control parameters.
Status Output	Error and status codes returned by API calls.

The module runs on a General Purpose Computing (GPC) platform. The GPC under test for this validation supports the following physical interfaces at the boundary of the GPC:

Table 4 - Description of GPC interfaces

Interface	FIPS 140-2 Logical Interface
Power Connector	Power Input
Power On/Off Button	Control Input
System Reset Button	Control Input
LEDs	Status Output
SATA	Data Input, Data Output
VGA	Status Output
USB	Data Input, Data Output
LAN	Control Input, Data Output, Status Output

5. MODES OF OPERATION

The module supports a FIPS Approved mode of operation and a non-Approved mode of operation. It is important to note that regardless of the mode of operation selected by the operator, the module will enforce the running of FIPS 140-2 required automatic power-up self-tests[1] and conditional self-tests. A mode must be selected during power-up initialization prior to using any service involving cryptography.

NOTE: It is mandated by the procedural guidance set forth in this Security Policy, that a transition from a FIPS Approved Mode of operation to the non-Approved mode of operation, or vice-versa, only take place if the following prerequisites have been met.

1. The operator must zeroize all plaintext CSPs.
2. The operator must issue a power-cycle.

Not following these procedures is an explicit violation of the Security Policy.

[1] Module implements a Default Entry Point as per FIPS 140-2 IG 9.10.

6. FIPS APPROVED MODE

This mode of operation limits access to APIs which only allow the use of FIPS Approved Security Functions. Attempting to use a non-Approved service from Table 11 while the module is in the FIPS Approved mode, will return error OSNCRYPTO_ERR_FIPS_NON_COMPLIANT.

To set the crypto module in FIPS Approved mode, the _osncrypto_set_library_mode() function needs to be called with the control input argument “OSN_CRYPTO_LIBRARY_FIPS_MODE”. The function will return “OSNCRYPTO_ERR_SUCCESS” to indicate the module has been successfully set to FIPS Approved mode. Please see Table 10 for a listing of all services available in this mode.

Sequence of events to place the module in the FIPS Approved Mode[2]:

1. Power on the module.
2. Issue API _osncrypto_is_self_test_success(); this shall return TRUE indicating all self-tests passed.
3. Issue API _osncrypto_set_library_mode() with the control input argument “OSN_CRYPTO_LIBRARY_FIPS_MODE”. The function will return “OSNCRYPTO_ERR_SUCCESS” to indicate the module has been successfully set to FIPS Approved mode.
4. Issue API _osncrypto_is_fips_approved(); this shall return TRUE indicating the module is in the FIPS Approved Mode.
5. Issue API _osncrypto_get_version(); the version of the module shall match the Security Policy.

[2] See section 5 prerequisites for zeroization and power-cycle if transitioning from a non-Approved mode of operation.

7. NON-APPROVED MODE

The non-Approved mode was implemented to allow additional APIs, which are otherwise blocked in FIPS Approved mode of operation. The additional APIs make use of non-approved cryptographic functions solely for backwards compatibility purposes (e.g. DES, MD5). Please see Table 10 and Table 11 for a listing of all services available in this mode.

Sequence of events to place the module in the non-Approved Mode[3]:

1. Power on the module.
2. Issue API _osncrypto_is_self_test_success(); this shall return TRUE indicating all self-tests passed.
3. Issue API _osncrypto_set_library_mode() with the control input argument "OSN_CRYPTO_LIBRARY_DEFAULT_NONFIPS_MODE". The function will return "OSNCRYPTO_ERR_SUCCESS" to indicate the module has been successfully set to non-Approved mode.
4. Issue API _osncrypto_is_fips_approved(); this shall return FALSE indicating the module is in the non-Approved Mode.
5. Issue API _osncrypto_get_version(); the version of the module shall match the Security Policy.

[3] See section 5 prerequisites for zeroization and power-cycle if transitioning from a FIPS Approved mode of operation.

8. ALGORITHMS

The module supports the following Approved Algorithms:

Table 5 - Approved Algorithms Table

CAVP Cert(s)	Algorithm	Standard	Mode	Key length (bits)	Use
C2096, C2098	AES	SP 800-38A	CBC	256	Encrypt/Decrypt
C2108, C2111	AES	SP 800-38F	KW	256	Encrypt/Decrypt
C2112, C2113	AES	SP 800-38E[4]	XTS	256	Encrypt/Decrypt
Vendor Affirmed	CKG	SP 800-133			Cryptographic Key Generation
C2106, C2109	DRBG	SP 800-90A	CTR_DRBG AES-256 DF	256	Deterministic Random Bit Generation
A810	HMAC	FIPS 198-1	HMAC-SHA-256	160-256	Message Authentication
C2108 C2111	KTS	SP 800-38F	KW	256	Encrypt/Decrypt
A809	PBKDF	SP 800-132[5]	HMAC-SHA-256	256	Password-based Key Derivation
C2097, C2099	SHS	FIPS 180-4	SHA-256 SHA-512		Message Digest

[4] Cryptographic protection of data on storage devices only (Module is a crypto library for storage applications).

[5] Key Derivation for storage applications only (Module is a crypto library for storage applications).

The module supports the following non-Approved but allowed algorithms:

Table 6 - Non-Approved but allowed algorithms

Algorithm	Caveat	Use
NDRNG	None	INTEL RDRAND used to seed approved DRBG. Minimum of 256-bits of security strength.

The module supports the following non-Approved algorithms in the non-Approved mode of operation[6]:

Table 7 - Non-Approved algorithms

Algorithm	Use
DES	Legacy decryption
MD5	Legacy digest verification

[6] The module must be set to “OSN_CRYPTO_LIBRARY_DEFAULT_NONFIPS_MODE” as per the procedure in section 7.

9. IDENTIFICATION AND AUTHENTICATION POLICY

The operator is defined as any consuming application that is linked to the OSNEXUS Crypto Library (*libosn_cryptolib.so.1.0.0, libcrypto.so.1.0.0 and libssl.so.1.0.0*). The module supports two roles, a Crypto Officer (CO) and a User role. Both roles have the same level of access to the module's services, hence role selection and assumption is implicit. The module does not support Authentication Mechanisms.

The Crypto Officer (CO) role is assumed by the human operator, physically present at the boundary of the GPC, performing the module installation. The installation is accomplished by the following sequence:

1. CO obtains the ISO install from OSNEXUS with the following binary images which after installation will be located at '/opt/osnexus/common/lib' :
    1. *libosn_cryptolib.so.1.0.0*
    2. *libcrypto.so.1.0.0*
    3. *libssl.so.1.0.0*

2. CO power-cycles the GPC and follows instructions to place the module into the FIPS Approved Mode as per section 6 above.

Table 8 - Roles and Authentication Mechanisms

Role	Authentication type	Authentication data
Crypto Officer (CO)	N/A	N/A
User	N/A	N/A

Table 9 - Strength of Authentication Mechanism

Authentication Mechanism	Strength of Mechanism
N/A	N/A

10. CRITICAL SECURITY PARAMETERS

The Critical Security Parameters (CSPs) used by the module are protected from unauthorized disclosure, modification, and substitution.

1. Data Encryption Keys XTS (Pool Keys)

Type: AES-XTS-256.
Generation: Supplied by caller from the unmodified output of the SP 800-90A DRBG via API _osncrypto_aes_generate_key(). As per SP 800-133 Section 7.1, key generation is performed as per the "Direct Generation" of Symmetric Keys which is an Approved key generation method.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: AES-XTS process.
Zeroization: CSP is actively overwritten via the Zeroization service and power-cycle.

2. DRBG Entropy Input

Type: 256-bits collected from NDRNG (INTEL RDRAND).
Generation: NDRNG.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: DRBG Key Generation process.
Zeroization: CSP is actively overwritten at the completion of the process and power-cycle.

3. DRBG Seed

Type: 384-bits of DRBG Seed Material (entropy_input, nonce, personalization_string).
Generation: SP 800-90A DRBG.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: DRBG Key Generation process.
Zeroization: CSP is actively overwritten when the module exits, the library destructor will call DRBG Uninstantiate function; will also be overwritten by power-cycle.

4. DRBG Internal State

Type: Internal state for SP800-90A AES-256-CTR DRBG with DF (K: 256 bits and V: 128 bits).
Generation: SP 800-90A DRBG.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: DRBG Key Generation process.
Zeroization: CSP is actively overwritten when the module exits, the library destructor will call DRBG Uninstantiate function; will also be overwritten by power-cycle.

5. Password for PBKDF2

Type: Password (20 character minimum in FIPS Mode enforced by the module).
Generation: N/A.
Establishment: N/A.
Entry: Plaintext.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: SP 800-132 PBKDF2 process.
Zeroization: Plaintext CSP is actively overwritten via the Zeroization service and power-cycle.

6. PBKDF2 Internal State

Type: SP 800-132 PBKDF2 (HMAC-SHA-256) Internal State and
Salt (generated by SP 800-90A DRBG).
Generation: SP800-132 PBKDF2 (HMAC-SHA-256) with a 1,000,000 iteration count.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: SP 800-132 PBKDF2 process.
Zeroization: CSP is actively overwritten at the completion of the process and power-cycle.

7. Key Encryption Key (KEK)

Type: SP 800-38F AES-256-KW KEK; used to support key wrapping/unwrapping for keys being stored in DISK. Consuming application must call _osncrypto_aes_wrapKey() to perform the key wrapping.
Generation: SP800-132 PBKDF2 (HMAC-SHA-256) with a 1,000,000 iteration count.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: AES-KW Process.
Zeroization: Plaintext CSP is actively overwritten via the Zeroization service and power-cycle.

8. Data Encryption Keys CBC

Type: AES-CBC-256 (Both the Key and IV are generated from the output of the SP 800-90A DRBG).
Generation: Supplied by caller from the unmodified output of the SP 800-90A DRBG via API _osncrypto_aes_generate_key(). As per SP 800-133 Section 7.1, key generation is performed as per the "Direct Generation" of Symmetric Keys which is an Approved key generation method.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: AES-CBC process.
Zeroization: Plaintext CSP is actively overwritten via the Zeroization service and power-cycle.

9. HMAC Key

Type: HMAC-SHA-256 Key (160 bits minimum)
Generation: Supplied by caller from the unmodified output of the SP 800-90A DRBG via API _osncrypto_generate_randomdata_buffer(). As per SP 800-133 Section 7.1, key generation is performed as per the "Direct Generation" of Symmetric Keys which is an Approved key generation method.
Establishment: N/A.
Entry: N/A.
Output: N/A.
Storage: RAM in plaintext.
Key-to-entity: HMAC process.
Zeroization: CSP is actively overwritten at the completion of the process and power-cycle.

11. ACCESS CONTROL POLICY

The module supports the following services in the FIPS Approved mode of operation:

Table 10 - Module services

Service Name	API(s)	CO	User
Set library mode	_osncrypto_set_library_mode	X	X
Symmetric key operations	_osncrypto_aes_generate_key _osncrypto_aes_encrypt _osncrypto_aes_encrypt_data _osncrypto_aes_decrypt _osncrypto_aes_decrypt_data _osncrypto_aes_wrapkey _osncrypto_aes_unwrapkey	X	X
Hashing operations	_osncrypto_generate_hash_sha512 _osncrypto_generate_hash_sha512_vec _osncrypto_generate_hash_sha256 _osncrypto_generate_hash_sha256_vec _osncrypto_generate_hash_sha256_list	X	X
HMAC Generation	_osncrypto_hmac	X	X
PBKDF2 operations	_osncrypto_pbkdf2_hmac	X	X
Random Number Generation	_osncrypto_generate_randomdata_buffer _osncrypto_generate_random_str	X	X
Zeroization	_osncrypto_cleanse_vector _osncrypto_secure_erase	X	X

Service Name	API(s)	CO	User
Data encoding operations	_osncrypto_base64_encode_string _osncrypto_base64_decode_string _osncrypto_base64_encode _osncrypto_base64_decode _osncrypto_vector2hex _osncrypto_plain2vector _osncrypto_hex2vector _osncrypto_hex2bin _osncrypto_bin2hex _osncrypto_intStr2LittleEnd128	X	X
Self-Test	N/A – automatically launched at power-up.	X	X
On-Demand Self-Test	_osncrypto_start_self_test	X	X
Show Status	_osncrypto_is_self_test_success _osncrypto_is_fips_error_mode _osncrypto_is_fips_approved _osncrypto_get_library_mode _osncrypto_get_lib_status _osncrypto_get_version _osncrypto_is_intel_cpu _osncrypto_is_amd_cpu	X	X

In addition to the services above, the module supports the following services **only** in the non-Approved mode of operation:

Table 11 – Module services available in Non-Approved mode

Service Name	API(s)	CO	User
DES Legacy Decryption	_osncrypto_des_decrypt _osncrypto_des_decrypt_data	X	X
MD5 Legacy Digest Verification	_osncrypto_generate_hash_md5 _osncrypto_generate_hash_md5_list _osncrypto_file_md5sum	X	X

The module enforces an Access Control Policy as to which services are allowed to access security relevant data. The types of access are listed in the following table.

Table 12 - Access type definitions

Access Type	Description
C	The item is created.
Z	The item is zeroized.
E	The item is read from memory and executed for a given operation.
W	The item is written or modified.

Service Name	CSPs	Access Type
Set library mode	N/A	N/A
Symmetric key operations	Data Encryption Keys XTS (Pool Keys) Key Encryption Key (KEK) Data Encryption Keys CBC	C, E, W E C, E, W
Hashing operations	N/A	N/A
HMAC Generation	HMAC Key	C, E, W, Z
PBKDF2 operations	Password for PBKDF2 PBKDF2 Internal State Key Encryption Key (KEK)	C, E, W C, E, W, Z C, W
Random Number Generation	DRBG Entropy Input DRBG Seed DRBG Internal State	C, E, Z C, E, Z C, E, W, Z

Service Name	CSPs	Access Type
Zeroization[7]	Data Encryption Keys XTS (Pool Keys) Password for PBKDF2 Key Encryption Key (KEK) Data Encryption Keys CBC	Z Z Z Z
Data encoding operations	N/A	N/A
Self-Test	N/A	N/A
On-Demand Self-Test	N/A	N/A
Show Status	N/A	N/A
DES Legacy Decryption	N/A	N/A
MD5 Legacy Digest Verification	N/A	N/A

[7] Other CSPs of the module are actively overwritten automatically at the completion of the service and by power-cycle: DRBG Entropy Input, DRBG Seed, DRBG Internal State, PBKDF2 Internal State, HMAC Key.

12. SECURITY RULES

The following is a list of security rules that support the FIPS Approved mode of operation and must be adhered to by the operator in order to comply with FIPS 140-2 requirements.

1. The OSNEXUS Crypto Library must be used as described in the Security Policy.
2. Installation of the module is the responsibility of the Crypto Officer (CO), the user is simply the consuming application invoking the library API. See section 9 for more details.
3. The module provides a FIPS 140-2 Approved mode of operation. Before the module can be used, it must be initialized as described in section 6.
4. The module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., for Business use).
5. The module inhibits data output during self-tests and error states. The data output interface is logically disconnected from the processes performing key generation and zeroization.
6. The zeroization service can be achieved by using the appropriate API functions *_osncrypto_cleanse_vector and _osncrypto_secure_erase.* Note, these functions must be called for each key or CSP object created. There is no one single function to zeroize all CSP objects. Additionally, once all CSPs have been zeroized, procedural guidance recommends the power-cycle of the module to complete the zeroization procedure.
7. In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP 800-133 (Vendor Affirmed). The resulting generated symmetric key is the unmodified output of the SP 800-90A DRBG.
8. When generating keys, the operator shall comply with requirements in SP 800-57 and generate a unique key per algorithm usage type (CBC, XTS, KW, HMAC).

9. Module complies with requirements in SP 800-132 for PBKDF2. Keys generated using PBKDF2 shall only be used in data storage applications. The minimum password length is 20 characters in FIPS Approved Mode of operation. When issuing the API _osncrypto_pbkdf2_hmac, the salt parameter must be set to "NULL". This action will enforce the internal generation of the salt by the module using SP 800-90A DRBG.
10. Module complies with requirements in SP800-38E and IG A.9 for AES-XTS. The module performs a key_1 vs. key_2 validation check to ensure the keys are not equal to one another. If they happen to fail this test, the module will enter the FIPS Error state. Similarly, the module also performs validations on the length of data units to ensure they cannot exceed 2^{20} AES blocks. A data unit length that exceeds this limit will transition the module to the FIPS Error state.
11. Module complies with requirements in SP 800-38A for AES-CBC IV construction, the modules generates the IV from the output of SP800-90A DRBG. The API "_osncrypto_aes_generate_key()" can be used to generate the AES-CBC key (256-bit) and the IV (128-bits) directly from the SP800-90A DRBG output.
12. Module complies with requirements in FIPS 198-1 for HMAC; in order to invoke the HMAC Generation service API _osncrypto_hmac() in a compliant manner with FIPS 140-2, the operator shall ensure the HMAC Key parameter is a key generated from the output of the module's SP 800-90A DRBG using API _osncrypto_generate_randomdata_buffer(). The module will enforce the key to be at minimum 160 bits, however it is recommended that the operator uses 256-bit keys for HMAC-SHA-256.

13. SELF-TESTS

The module supports three forms of self-tests: power-on, on-demand, and conditional self-tests. The power-on self-tests are mandatory and run automatically without operator intervention. The on-demand self-tests can be invoked by the operator anytime by issuing the API _osncrypto_start_self_test (Note: this will perform the identical set of self-tests issued during power-on including the integrity check, if any error is encountered module transitions to the FIPS Error state).

All data output is inhibited during the self-test process. If any self-test fails, the module will enter the FIPS Error state and exit the process. The module supports a single FIPS Error state (OSNCRYPTO_LIBRARY_STATUS_FIPS_ERROR) and depending on the error the module can provide an exit status code. If the self-tests pass successfully, the module is now operational and will accept incoming API Calls. For both cases, failure or success, the operator can explicitly query the module via the following APIs to determine the state of the module:

Table 13 - Status indicators for Self-Tests

Self-Test Status	Show status APIs and expected returns
Self-Tests Passed	1. Issue API _osncrypto_is_self_test_success() Return: **TRUE** 2. Issue API _osncrypto_is_fips_error_mode() Return: **FALSE**
Self-Tests Failed	1. Issue API _osncrypto_is_self_test_success() Return: **FALSE** 2. Issue API _osncrypto_is_fips_error_mode() Return: **TRUE**
	Failure of any power-up test will return Exit code: *2* Failure of SP 800-90A Key Generation at run-time will return Exit code: *5* Failure of an NDRNG or DRBG conditional test will return Exit code: *6* Failure of an XTS IG A.9 key validation will return Exit code: *3* Failure of an XTS SP 800-38E data unit validation will return Exit code: *4*

The module supports the following self-tests:

Power-up

1. HMAC-SHA-256 Integrity Test
2. AES-256 Key Wrap KAT
3. AES-256 Key Unwrap KAT
4. AES-256-CBC Encrypt KAT
5. AES-256-CBC Decrypt KAT
6. AES-256-XTS Encrypt KAT
7. AES-256-XTS Decrypt KAT
8. AES-256-CTR with DF DRBG KAT
9. AES-256-CTR with DF DRBG SP 800-90A Section 11.3 Health Tests
10. HMAC-SHA-256 Generation/Verification KAT
11. SHA-256 KAT
12. SHA-512 KAT
13. PBKDF2 HMAC-SHA-256 KAT

Conditional

1. NDRNG (Intel RDRAND) Continuous Random Number Generation Test (32-byte)
2. AES-256-CTR with DF DRBG Continuous Random Number Generation Test (16-byte)
3. Pairwise consistency test: N/A
4. Software load test: N/A
5. Manual key entry test: N/A
6. Bypass test: N/A
7. Other conditional tests:
    - AES-XTS IG A.9 KEY_1 != KEY_2 validation
    - AES-XTS SP 800-38E length of data units shall not exceed 2^{20} AES blocks

14. PHYSICAL SECURITY POLICY

The module is implemented as a software-only module, and as such the physical security section of FIPS 140-2 is not applicable.

Table 14 - Physical Security Mechanisms

Physical security mechanisms	Recommended frequency of inspection/test	Inspection/test guidance details
N/A	N/A	N/A

15. MITIGATION OF OTHER ATTACKS POLICY

The Mitigation of Other attacks security section of FIPS 140-2 is not applicable to the module. The module is not designed to mitigate against attacks outside the scope of FIPS 140-2.

Table 15 – Mitigation of other attacks

Other attacks	Mitigation mechanism	Specific Limitations
N/A	N/A	N/A